



SOURCEPRO[®] 2016 RELEASE NOTES



Introduction

In this Document

These release notes contain a summary of new features and enhancements, late-breaking product issues, migration from earlier releases, and bug fixes. The information relating to the most recent release appears first in this document. Following sections contain release notes for all relevant prior releases. The notes for all prior releases are applicable to the most recent release as well.

For change logs or other information on previous releases, please contact Technical Support.

NOTE » Rogue Wave products may contain undocumented interfaces. These interfaces are not supported for general use and may be changed or removed from release to release.

NOTE » The version of this document in the product distribution is a snapshot at the time the product distribution was created. Additional information may be added after that time because of issues found during distribution testing or after the product is released. To be sure you have the most up-to-date information, see the version of this document on the Rogue Wave web site:
<http://www.roguewave.com/support/product-documentation/sourcepro.aspx>.



SourcePro 2016

New Features or Enhancements

Introduction

This section details changes from SourcePro 13.2 to SourcePro 2016.

Updated Platform Support

Specific platform information is not included in this document. Please see the Support Matrix at <http://www.roguewave.com/products/sourcepro.aspx>. Select *Learn about supported platforms*. Note that the support matrix document displays **bold** text for any updates added in the current release.

General

- The compilation time of all SourcePro libraries was improved by optimizing string concatenation. (SP-14473)
- Single-threaded and Solaris thread builds are no longer supported. All modules are built with the WIN32 thread library on Windows, and the POSIX thread library on all other platforms. (SP-22390)

SourcePro Core

Essential Tools Module

- New classes `RWTimeTuple` and `RWTimeTupleOffset` have been introduced to support manipulation of date/time information similar to a `struct tm`. (SP-20987)
- New classes `RWTimeTupleFormatter` and `RWTimeTupleOffsetFormatter` have been introduced to support converting the new `RWTimeTuple` and `RWTimeTupleOffset` to string representation. (SP-21619)
- `rw_slist` has been reimplemented to improve performance. This also improves the performance of `RWPtrSlist` and `RWValSlist` which are implemented in terms of `rw_slist`. (SP-20036)
- `rw_hashmap`, `rw_hashmultimap`, `rw_hashmultiset` and `rw_hashset` have been updated to take advantage of the new `rw_slist` implementation. (SP-20036)
- `RWAnsiLocale` has been updated to no longer make unnecessary copies of `std::locale`. (SP-20438)
- `RWBTreeDictionary`, `RWHashDictionary` and `RWIdentityDictionary` now implement `insert`, which accepts an `RWCollectableAssociation` and inserts the associated key and value. (SP-22389)
- The performance of `entries()` and `size()` on all hash-based containers was improved. (SP-21535)

SourcePro DB

DB Interface Module

- `RWTHash` has been specialized for `RWDBBlob`, `RWDBDuration`, `RWDBMBString` and `RWDBValue` in order to simplify the use of those classes in hash-based containers. (SP-21288)
- `RWDBValue::hash()` has been updated to leverage `RWTHash` for the contained type in order to provide a better hash distribution. (SP-21288)
- `RWDBValue` has been updated to store instances of `RWTimeTuple` and `RWTimeTupleOffset`. (SP-20284)
- `RWTimeTuple` support added to `RWDBTBuffer`, `RWDBBuffer`, `RWDBTracer`, `RWDBOSql`, `RWDBBulkInserter`, and `RWDBBulkReader`. (SP-21208)

DB Access Module for DB2 CLI

- The performance when reading wide-character data on Windows has been improved. (SP-20274)

Bug Fixes

Introduction

This section details changes from SourcePro 13.2 to SourcePro 2016.

SourcePro Core

Essential Tools Module

- `rw_slist::unique` no longer exits abnormally when called on an empty list. (SP-20036)
- `RWDate(struct tm*)` no longer fails if the `struct tm` contains a `tm_year` between -1900 and -1801. (SP-21146)
- `RWDateTime::asRWDate()` no longer fails when creating an `RWDate` if the `RWDateTime` is in the years between 0 and 99. (SP-15538)
- `RWDateTime::compareTo()` no longer throws an exception if one or both of the arguments represent sentinel values. This change brings the behavior of `RWDateTime::compareTo()` into alignment with `operator<` and other relational operators. (SP-15540)
- `RWBasicUString(RWSize_T)` no longer asserts in debug builds when the input parameter is equal to 0 (SP-20849)
- `RWLocaleSnapshot` was optimized to remove an unnecessary memory allocation. (SP-20437)
- `RWStandardWString::toAscii()` was optimized to remove an unnecessary memory allocation. (SP-20440)
- `operator+=` no longer results in a double-delete when used with collections of type `RWBTreeDictionary`, `RWHashDictionary` and `RWIdentityDictionary`. (SP-22389)
- `operator>>` no longer produces an invalid pointer when extracting objects that inherit from `RWCollectable` and another type. (SP-22828)
- Fixed memory in use errors in `getRWNilCollectable()` and `RWBasicUString::getNullDeallocator()` by switching from dynamic to static storage. (SP-20478)
- `RWLocaleSnapshot::asString` no longer generates invalid locale-dependent string representations (`x`, `c`) on Windows when the underlying locale-dependent functions use a non-gregorian calendar. (SP-22763)
- `RWLocaleSnapshot::asString` now generates locale-dependent time representations (`X`) using correct AM/PM specifiers and with an appropriate 12-hour or 24-hour clock. (SP-23017)

Advanced Tools Module

- Fixed a race condition in the destructor for `RWSymbol`. (SP-20460)

Threads Module

- A header guard conflict between `rw/functor/fwd.h` and `rw/functor/list/fwd.h` was resolved. (SP-18506)

SourcePro DB

DB Interface Module

- `RWDBValue::compareTo()` now returns the correct result when comparing `RWDBValue::UString` values. (SP-14593)
- `RWDBValue::asBlob()` now returns consistent results for `RWDBValue::Date` and `RWDBValue::DateTime` types. (SP-21306)
- `RWDBDuration::asString()` now adds 's' after the unit for plural negative values. (SP-21292)

DB Access Module for MySQL

- `MYSQL_TYPE_TINY` data is now correctly treated as an integral type. (SP-22068)

DB Access Module for Oracle

- `RWDBOCISystemHandle` now properly cleans up connection when an exception is thrown from a connection callback. (SP-20467)

SourcePro Analysis

Linear Algebra Module

- All memory allocations in the Linear Algebra Module were updated to align memory on 16-byte boundaries when linked against the Sun Performance Library on Solaris x86. (SP-17965)
- Fixed memory in use errors when building exception strings by switching from dynamic to static storage. (SP-20471)

Currency Module

- Fixed memory in use errors in an internal class used by `RWDecimal` by switching from dynamic to static storage. (SP-20473)
- Fixed memory in use error in `format2` example by switching from dynamic to local storage. (SP-20473)

SourcePro Net

Secure Communications Module

- Updated `RWSecureSocketPackageInit::seedRNGFromFile()` and `RWSecureSocketContext::loadVerifyLocations()` to avoid an invalid memory access in `stat()` on Windows. (SP-20476)

Migration Guide

Introduction

This section details API changes from SourcePro 13.1 to SourcePro 2016.

NOTE » Throughout this document, a class member may be described as "undocumented". This means that its use is not described anywhere in Rogue Wave documentation, and it is not public. Using undocumented APIs is not recommended or supported.

SourcePro Analysis

Intel Math Kernel Library

SourcePro Analysis no longer ships with Intel's Math Kernel Library. For information on options for downloading the Math Kernel Library from Intel, see: http://software.intel.com/en-us/articles/free_mkl.

SourcePro Core

Essential Tools Module

ISO8601 Compatibility Parser was deprecated. (SP-21129)

The enumerator `RWDateTime::iso8601_compat` was deprecated. Applications should use one of the other `RWDateTime::Format` enumerators instead.

RWLocaleSnapshot::refresh() was removed. (SP-20437)

The function `RWLocaleSnapshot::refresh()` was removed. Applications that need custom locale definitions should generate them using the tools provided with the operating system.

Threads Module

Package-level tracing was removed. (SP-17826)

The package-level tracing option was removed from the RCB GUI and package-level tracing was removed from all SourcePro packages. Configuring user-level tracing is still available from the RCB GUI.

SourcePro DB

DB Interface Module

RWDBValue::compareTo() return value specification was expanded. (SP-21296)

The return values for `RWDBValue::compareTo()` were overspecified, implying that only the values `-1`, `0` and `1` should be expected. This was incorrect; a less than relationship will return a value less than `0`, and a greater than relationship will return a value greater than `0`. The documentation has been updated to reflect the actual behavior of `RWDBValue::compareTo()`. User code that is defined to check for a specific value other than `0` should be updated to instead check the value relative to `0`.

SourcePro Net

Secure Communication Module

Important OpenSSL Security Migration (SP-22901)

By default, OpenSSL 1.0.2h has disabled the `SSLv2` protocol, along with various ciphers from multiple protocols that are considered unsafe. An exception is thrown when an `RWSecureSocketMethod` is constructed with `RWSecureSocketMethod::SSLv2`, `RWSecureSocketMethod::SSLv2Client`, or `RWSecureSocketMethod::SSLv2Server`. User code should use one of the currently supported protocols documented in the `RWSecureSocketMethod` section of the reference guide. For more information about OpenSSL's removal of `SSLv2` see: <https://www.openssl.org/news/changelog.html>

Dependencies, Known Issues, and Limitations

Installation

General Installation Issues

NOTE » Each SourcePro release is a replacement for, not an upgrade to, the previous version. Always perform an installation into a new area, and not into the same install directory as a previous version.

OS-specific Issues

Windows

- **The user needs to have administrator privileges to run the installer.** If you don't have administrator privileges, the installer will prompt you to enter the administrator's password.
- **UNC installation paths not supported.** For network installs on Windows, using a UNC path to specify the installation location is not supported. The problem is actually in RCB which, in such installs, is unable to create valid makefiles. So instead, create a mapped network drive to the desired network location. For the mapped network drive, be sure to check "Reconnect at Logon" checkbox. If this option is not selected, the mapped network drive will not appear when browsing for an install location.

Solaris

- **UTF-8 locale results in erroneous text boxes that may obscure GUI.**

On Solaris, when invoking the installer with the **LANG** environment variable set to a UTF-8 locale (i.e., **en_US.UTF-8**), a series of small boxes with the word "iimx" and/or "English/European" may be generated while navigating through the installer, possibly obscuring portions of the GUI. These boxes can be suppressed by unsetting the **LANG** variable or setting it to another locale (e.g., **C**, **POSIX**, **en_US.ISO8859-1**).

RCB

- **Under specific build constraints, RCB menu options all become disabled**

Given certain rare build constraints, all RCB menu options become disabled (grayed out) due to constraint checking. If this occurs, exiting the screen with constraint checking enabled is not possible. If you wish to go back to alter previous screens, first turn off constraint checking, move to the previous screen, and then re-enable constraint checking. This situation is triggered when building both the Internationalization library and any library that requires multithreading, while simultaneously not building the Threads library.

- **Missing Top-Level Build Results HTML Page**

Whenever you invoke RCB to build some set of libraries, there should be a build results HTML page at `<buildspace>/records/results/index.html`. However, this occurs only the first time you invoke RCB; for subsequent builds, this file is missing. You can, however, find the same information by looking in the subdirectory for the buildtype of the build just performed. For example, the results for a 12d build would be found at `<buildspace>/records/results/12d/index.html`. (SP-15699)

- **LANG Environment Variable Setting on Solaris**

When invoking the RCB GUI with the `LANG` environment variable set to a UTF-8 locale (i.e., `en_US.UTF-8`), a series of small boxes with the word 'iimx' on them may be generated as the user navigates through the RCB GUI pages. These boxes may obscure portions of the GUI. These boxes can be suppressed by unsetting the `LANG` variable or setting it to another locale (e.g., `C`, `POSIX`, or `en_US.ISO8859-1`). (INF-4361)

- **Launching multiple RCB instances on Windows results in an error**

Attempting to launch multiple RCB instances in Windows may result in an error `The process cannot access the file because it is being used by another process`. If this occurs, start only one instance of RCB at a time. (SP-15715)

- **UNC Path Incompatibility**

On Windows, UNC (Universal Naming Convention) paths generally are exports provided by remote file servers. These paths take the form `\\host\share\path`. While it is possible to launch RCB from a UNC location, all paths specified in the buildspec must be located on paths identified using a drive letter.

There are three cases where the UNC path may cause problems:

- You export the build and the local buildspace, the export buildspace, or both are UNC paths. The result is a fatal error referring to a file `null`. The stack trace in the RCB `err.txt` file will reference a `java.io.IOException: Invalid argument thrown from java.io.WinNTFileSystem`.
- If you create a local buildspace, you can create the buildspec without issue, but the characterization process will fail. The resulting error message indicates that null is not recognized as an internal or external command, operable program or batch file.
- If a third-party library uses a UNC path, RCB appears to handle these paths correctly. However, when RCB attempts to link generated libraries and examples against libraries located on UNC paths, the link operation reports undefined symbols.

Mapping the UNC paths and using the mapped drives' letter in your RCB build works around all three of these issues. (SP-15706)

- **No spaces in #include path strings**

RCB-generated makefiles do not put quotes around include path strings. If the path to the installation directory (local buildspace), export buildspace, or a user-provided third party library contains spaces, then compile failures will result. One way to prevent these failures is to ensure that the paths for the locations in question don't contain spaces. However, this may not be possible in some environments. If it isn't possible to remove all spaces from the paths, select the build action of "just generate makefiles" and then perform the build through RCB. Once RCB has generated the required makefiles, alter the makefiles to quote the

path(s) which contain spaces, then execute the relevant `makeall_ <buildtype>` script (located in the root of the buildspace) to perform your build manually. (SP-15689)

SourcePro Core

Essential Tools Module

- **Using the Solaris SunPro Compiler and the native C++ Standard Library (SP-9247)**

When using the Solaris SunPro compiler and the native C++ Standard Library, the number of bytes counted by an `RWAuditStreamBuffer` cannot be printed without forcing the compiler to invoke the member conversion operator. The following code demonstrates the issue:

```
RWAuditStreamBuffer buf;  
RWbostream bcount(&buf);  
std::cout << buf << std::endl;
```

This problem affects the Essential Tools examples `manual/auditbuf` and `manual/auditbufx`. The problem is not observed when using the STLPort C++ Standard Library.

- **On HP-UX, using other than C Locales with the C++ Standard Library**

The C++ Standard Library that comes with the HP-UX C++ compiler has problems with locales other than the "C" locale. This can cause failures in `RWAnsiLocale::stringToDate()`. (SP-10843 / HP CR# QXCR1000990079)

Internationalization Module

Dependencies

The Internationalization Module depends on the International Components for Unicode (ICU), a set of open-source C and C++ libraries developed and maintained by IBM. The ICU is included in binary form with the Internationalization Module. When building the Internationalization Module via RCB, you may use the provided binary (by default), or an installation of ICU already existing on your system.

The installer always copies the ICU onto your disk if you choose to install the Internationalization Module. You may simply remove it if it is not needed.

Unicode Regular Expression Support

Unicode Regular Expression support, via *RWURegularExpression*, is currently supported at Levels 1, 2 and 3 as presented in version 5.1 of the Unicode Technical Report number 18. This technical report contains suggested features and syntax for a Unicode Regular Expression solution. The technical report is available online at:

<http://www.unicode.org/reports/tr18/tr18-5.1.html>.

At Level 1, string character comparisons are performed based on the bitpatterns of the characters in the pattern and search strings; canonical equivalents are not matched. No locale-dependent features are currently supported, with the exception of the detection of breaks in the text (character, word, line, and sentence breaks). In the Internationalization Module, this level is called "Basic Conformance Level".

At levels 2 and 3, character comparisons are done with locale-specific knowledge, canonically equivalent character sequences match, and surrogate pairs are correctly handled. In the Internationalization Module, this level is called "Tailored Conformance Level".

Please note that, although version 5.1 of UTR 18 was used to direct the development of *RWURegularExpression*, UTR 18 has since been updated. The latest version of this technical report is changed in two notable ways:

- First, direct support for surrogates is now recommended at Level 1, whereas it had previously been recommended for Level 2 and higher.
- Second, the suggested syntax for the construction of Unicode Regular Expression patterns has been changed to a more PERL-like syntax.

At this time, *RWURegularExpression* conforms to the earlier version of UTR 18.

Threads

Threading Package

- The *Threads Platform Guide* documents that user-defined stack attributes are supported on Linux. This is not currently the case. We continue to investigate providing stack attribute support on this platform; however, we are unable to do so at this time. (SP-13419)

Smart Pointers Package

- The *RWOnlyPointer* copy constructor and assignment operator take `const RWOnlyPointer` references, but casts away `const` on the parameter in order to invalidate it.

The parameters should be changed to non-`const` references to be correct, but this could potentially break user code which depends on being able to pass a `const` parameter. This issue will be fixed in the future.

Do not use *RWOnlyPointer* instances in containers that expect assignability and copy constructability. This could cause issues because the value passed to the copy constructor or assignment operator will be invalidated. (SP-10222)

Solaris 11 Threading Issue

- Oracle Solaris 11 changed the behavior of various functions related to setting and retrieving thread attributes. These changes are incompatible with assumptions the SourcePro Threads Module makes and may lead to unexpected exceptions, silent failures, or other errors. For this release, it is recommended that users avoid thread attributes related to priority, scheduling or inheritance on Solaris 11. This behavior will be reviewed in a future release.

SourcePro DB

DB Access Module for DB2 CLI

DB2 10.5 fixpack 5 on Solaris

- The DB2 client library is unable to load the socket library on Solaris. When this happens the database returns "SQL1042C An unexpected system error occurred. SQLSTATE=58004". The workaround is to add '-lsocket' to the link line of the executable. (SP-21466)

DB Access Module for MySQL

- **RWDBDatabase and multithreaded applications.**

In multithreaded applications, producing *RWDBDatabase* objects using an `RWDBManager::database()` call in each thread simultaneously results in a segmentation fault. This is due to a MySQL native call that is not multithread-safe.

To work around this problem, serialize the production of the first *RWDBDatabase* object. Alternatively, call MySQL native API `mysql_server_init()` (`mysql_library_init()` for MySQL 5.x) before accessing the DB Access Module for MySQL. (SP-8553)

DB Access Module for ODBC

DB2 Driver

On UNIX platforms, with the unixODBC 2.3.2 driver manager and the DB2 9.7 FixPack 7 client:

- The following `odbcinst.ini` options were used with the DB2 ODBC driver:

```
DontDLClose = 1
FileUsage = 1
Threading = 0
```

The threading option may address possible `SIGABT` crashes that have been observed in multithreaded ODBC applications.

- The following `db2cli.ini` options were used with the DB2 ODBC driver:

```
LongDataCompat=1
MapGraphicDescribe=2
```

Microsoft SQL Server Driver

- This driver ignores the database argument passed to the *RWDBManager*'s `database()` function. Instead, the database associated with the `RWDBDatabase` object returned by the `database()` function is the default database of the user.

Informix Driver

- **Return values from stored procedures returned as result sets.**

ODBC allows the driver implementer to decide how to handle return values from stored procedures. A common convention is to support a bound return value. In this scenario, SourcePro DB allows retrieving the return value in the following manner:

```
RWDBDatabase db = ...
RWDBConnection conn = db.getConnection();
RWDBStoredProc sproc = db.storedProc("_procedurename_", conn);

// execute the stored procedure
sproc.execute(conn);

// retrieve the return value
RWDBValue val = sproc.returnValue();
```

The Informix ODBC driver supports multiple return values, and as a result follows an alternate convention, in which the return values are returned as a result set for the function. In that scenario, the return value can be retrieved from the result set, like so:

```
// execute the stored procedure
RWDBResult res = sproc.execute(conn);

// create a reader for the result set.
RWDBReader reader = res.table().reader();
RWDBValue val;

reader() >> val; // retrieve the return value
```

This behavior is consistent with the legacy Informix Access Module.

- **Calling stored procedures with no parameters.**

ODBC defines the call semantics for a stored procedure with no parameters as:

```
{ call procedurename }
```

Informix does not support this syntax and instead uses an alternate syntax in this scenario:

```
{ call procedurename() }
```

The Informix syntax is interpreted by conforming ODBC implementations as calling the procedure `procedurename` with one argument, where that argument has a default value.

The ODBC Access Module's *RWDBStoredProcedure* implementation uses the first calling convention when invoking a stored procedure with no parameters. In order to call an Informix stored procedure with no parameters, use the *RWDBOSql* interface instead. For example:

```
RWDBDatabase db = /* ... */  
RWDBConnection conn = db.getConnection();  
  
const RWCString callStmt = "{ call procedurename() }";  
RWDBOSql osql(callStmt, RWDBOSql::Procedure);  
  
osql.execute(conn);
```

- **Dropping columns.**

The SQL syntax for dropping a column allows for an optional **COLUMN** keyword. The syntax is:

```
ALTER TABLE <table> DROP [COLUMN] <column>
```

Some databases require the optional **COLUMN** keyword, others allow it, and Informix prohibits it. The ODBC Access Module emits the **COLUMN** keyword, causing a conflict when connecting to Informix. To resolve this, use the *RWDBOSql* interface. For example:

```
RWDBDatabase db = /* ... */  
  
RWDBTable table = db.table("table");  
// ...  
  
RWDBColumn col;  
col.type(RWDBValue::Int).name("col");  
  
// ...  
table.dropColumn(col);
```

This last line could be rewritten to use *RWDBOSql* as:

```
char buf [1024];  
snprintf(buf, sizeof buf, "ALTER TABLE %s DROP %s",
```

```

        table.name().data(), col.name().data());
RWDBOSql osql(buf);

osql.execute(con);

```

- **Operations involving a TEXT column.**

If you use `RWDBBulkInserter` to insert an array of strings with a length of 255 or less into a `TEXT` column, the insertion will fail and the Informix Database will return the error:

```
HY0000 Illegal attempt to convert Text/Byte blob type.
```

Resolve this by specifying the width parameter while constructing the `RWDBTBuffer` that is bound against the `TEXT` column. For example:

```

// Bulk inserting 100 rows of strings with width of 256
RWDBTBuffer<RWCString> buff(100, 256);
buff[0] = "abc";

// ...

```

If you encounter the same Informix database error from any other operation, use the `RWDBOSql` interface along with `RWDBTBuffer`. For example:

```

RWDBDatabase db = /* ... */

RWDBInserter inserter = db.inserter("table");

RWCString str1;
RWDBBoundExpr expr1(&str1, &nullIndicator1);

inserter << expr1;
str1 = "abc";

inserter.execute();

```

Here's an example using `RWDBOSql` and specifying the width parameter of the `RWDBTBuffer` with the SQL statement obtained by using the `RWDBTracer` interface:

```

RWDBDatabase db = /* ... */

RWDBOSql sql("INSERT INTO table VALUES (?");
RWDBTBuffer<RWCString> str1(1, 256);

sql << str1;

```

```
str1[0] = "abc";
```

```
sql.execute();
```

- **Clustered indexes.**

The SQL syntax for creating a clustered index is slightly different for Informix than other databases. The normal syntax is:

```
CREATE UNIQUE CLUSTERED INDEX <index> ON <table>(<column>)
```

Informix uses **CLUSTER** instead of **CLUSTERED**. The ODBC Access Module emits **CLUSTERED**, causing a conflict when connecting to Informix. Resolve this using the *RWDBOSql* interface. For example...

```
RWDBDatabase db = /* ... */

RWDBTable table = db.table("table");

// ...

RWDBColumn col;
col.type(RWDBValue::Int).name("col");

// ...

table.createIndex("index", schema, conn, true, true);
```

Rewrite the last line to use *RWDBOSql* as:

```
char buf [1024];

snprintf(buf, sizeof buf, "CREATE UNIQUE CLUSTERED INDEX %s ON %s(%s)",
         "index", table.name().data(), col.name().data());

RWDBOSql osql(buf);

osql.execute(con);
```

- **Inserting multibyte string literals not supported.**

Informix does not correctly handle inserting multibyte string literal values. Attempted insertions may result in either data corruption or an application crash, in some cases. Inserting multibyte strings via bound values works correctly. (SP-16804, IT02280)

- **Incorrect value returned by *RWDBOSql::rowsAffected()* when inserting with data-at-execution.**

RWDBOSql::rowsAffected() may return an incorrect value when a row is inserted using data-at-execution. Inserting a single row at a time avoids this issue. (SP-16579, IT02184)

- **Reading BIGINT values into a long long type is not portable.**

On some platforms, reading **BIGINT** data in to a **long long** datatype may result in the return of an incorrect value in which the upper and lower 32-bit values are swapped in the resulting output. This issue was observed with 32-bit Solaris x86 builds, but may affect other platforms as well. To ensure portability, it is recommended that values not be read in to the **long long** or **unsigned long long** datatypes. (SP-17188, IT02278)

- **Unable to describe default values for columns in a table schema.**

Attempting to describe a column that has a default value may result in incorrect or incomplete information. This issue affects only describing a column with a default value, creating a column with a default value and leveraging default values when inserting data are unaffected. (SP-16582, IT02230)

- **Binding 64-bit integral types not supported with input parameters.**

Binding a 64-bit integral type as an input parameter results in incorrect data being passed to the database. This affects only bound values; literal values are unaffected. (SP-17293, IT02552)

- **Trailing whitespace removed when inserting into a VARCHAR column.**

When strings are inserted into a **VARCHAR** column, trailing whitespace is removed. This behavior is not observed when inserting strings into a **TEXT** column. (SP-17187, IT01262)

- **Unable to set login timeout on Linux.**

On Linux, setting the login timeout (**RWDBODBCLibEnvironmentHandle::timeout**) has no effect. This is caused by a conflict in the handling of this value between unixODBC and the Informix driver. (SP-17187, IT02279)

- **Delimited names not supported.**

Informix requires delimited table names to be quoted when passed to ODBC API functions such as **SQLColumns** and **SQLProcedures**. This conflicts with the ODBC Access Module and other ODBC drivers that expect those names to be unquoted. Due to this limitation, delimited names are not currently supported with Informix. (SP-16743, IT01153)

DB Access Module for PostgreSQL

- **HP-UX Itanium and multithreaded builds.**

Multithreaded builds of the DB Access Module for PostgreSQL on HP Itanium must be built using a thread-enabled PostgreSQL client, which can be built from PostgreSQL source using the configure option **--enable-thread-safety**. (SP-12322)

- **PostgreSQL on HP-UX.**

On HP-UX, to avoid a conflict between the X/Open networking API and the BSD networking API, PostgreSQL must be compiled to use the alternative X/Open symbol names.

While configuring PostgreSQL on HP-UX, the following compiler flags should be specified through the **CFLAGS** argument:

-D_XOPEN_SOURCE=600 -D_HPUX_ALT_XOPEN_SOCKET_API

DB Access Module for Sybase

- **Binding 8-byte long and unsigned long data.**

The Sybase Client-Library `CS_INT` datatype is strictly 4 bytes, even on 64-bit platforms. Binding a `long` or `unsigned long` on a platform with an 8-byte `long` results in data truncation. Use the `int` datatype instead. (SP-14608)

- **Cursor Limitation.**

This library has problems handling decimal columns with `NULL` values through cursors. On some platforms, deleting a row with null values in decimal columns through a cursor causes a core dump. In some platforms, it returns non-null values for `NULL` columns. Sybase has identified this as a bug. (Sybase #94310 / SP-7765)

- **RWDBOSql limitation for RWCString data.**

If a user attempts to bind `RWCString` data having length greater than 16384 bytes into a column of type `TEXT`, the data gets truncated to 16384 bytes. This is due to the page-size limitation in Sybase.

A workaround is to insert the data as a literal rather than a placeholder. All DB Data Manipulation Classes will put `RWCString` data greater than 16384 bytes as literal in the SQL sent for execution. However, while using class `RWDBOSql`, the user must perform this before passing the statement to the `RWDBOSql` instance. (Sybase case #10832821 / SP-7789)

Bulk Insertion

The DB Access Module for Sybase supports bulk writing through the array interface provided by the Sybase Bulk-Library function `blk_bind()`, which is part of the Open Client and Open Server common libraries. However, Sybase Bulk-Library does not support insertion with column-list. Here are some other specifics on using bulk insertion with SourcePro DB and Sybase:

- The `RWDBBulkInserter` associated with the DB Access Module for Sybase can accept `RWDBTBuffer<RWDBBlob>` with elements up to 64K in width. Elements of width greater than 64K may corrupt the database.
- Bulk insertion is implemented based on the Sybase Bulk-Library, which may impact the recoverability of the database in case of failures. See the Sybase documentation on "Bulk-Library client programming" for full details.
- The Sybase Bulk-Library call `blk_bind()` doesn't report errors when `NULLS` are inserted into a column that doesn't accept `NULL` values.

OS-Specific Issues

Solaris

Executables using Sybase Access Module multithreaded dynamic builds on Solaris operating systems have been observed to produce intermittent segmentation faults if not linked with Sybase vendor libraries. To work around this issue, Sybase vendor libraries or the Sybase Access Module library can be linked to the executable. The Sybase Access Module examples link the Sybase Access Module library. As these failures are intermittent, if you do not observe these failures, you do not need to link these libraries to your executables. (SP-9472)

SUSE Linux Enterprise Server 11

Executables using Sybase Access Module multithreaded dynamic libraries on SUSE Linux Enterprise Server need to link the Sybase vendor libraries. Not linking the Sybase vendor libraries to the executable has been observed to produce intermittent segmentation faults. The Sybase vendor libraries can be linked directly or by linking the Sybase Access Module library to the executable. The Sybase Access module examples link the Sybase Access Module library.

As these failures are intermittent, if you do not observe these failures, you do not need to link these libraries to your executables. (SP-8505/RW# 32062)

Red Hat Enterprise Linux 5

Executables using Sybase Access Module multithreaded dynamic libraries on Red Hat Enterprise Linux 5 need to link the Sybase vendor libraries. Not linking the Sybase vendor libraries to the executable has been observed to produce intermittent segmentation faults. The Sybase vendor libraries can be linked directly or by linking the Sybase Access Module library to the executable. The Sybase Access Module examples link the Sybase Access Module library. As these failures are intermittent, if you do not observe these failures, you do not need to link these libraries to your executables. (SP-9426)

DB Access Module for Oracle OCI

- **Oracle Heterogeneous Services.**

When using Oracle Heterogeneous Services for SQL execution, you need to use explicit Transaction Management using `beginTransaction()` and `commitTransaction()`. By default, this module uses `OCI_COMMIT_ON_SUCCESS`, whereas heterogeneous services require transactions.

- **Oracle and LONG columns.**

When a result set fetched from Oracle has a `LONG` column in it and no buffer is bound for the `LONG` column, then Oracle returns either `ORA-03106: fatal two-task communication protocol` or `ORA-00127 dispatcher s does not exist`. (Oracle bug #4919068 / SP-8671)

- **Breaking out of polling operations with OCIBreak.**

Calling `OCIBreak` to terminate polling operations doesn't work. This results in ORA-03127 when the next operation is attempted on the connection. `OCIBreak` is called when:

- Reading into a *RWDBTBuffer*<char> when the buffer is not large enough to hold the data.
- Returning false or throwing an exception from *RWDBDataCallback::onFetch()*. This has been identified as Oracle bug #10606993.

• Using SYS.XMLTYPE datatype.

The Oracle datatypes *SYS.XMLTYPE* is supported by SourcePro DB, but extra steps are required in some cases. When inserting data using *RWDBInserter*, to use the best transfer mechanism the DB Interface Module needs to know that the underlying table has a *SYS.XMLTYPE* column. The easiest way to do this is by calling the *RWDBTable::fetchSchema()* method before creating your *RWDBInserter* instance. Here's an example that demonstrates this:

```
// CREATE TABLE foo(c NUMBER(9), xmlcol SYS.XMLTYPE);
RWDBTable foo = myDb.table("foo");
foo.fetchSchema(myConn);

RWDBInserter ins = foo.inserter();
int c;
RWCString xml;

// fill in data ...
ins << c << xml;
ins.execute(myConn);
```

OS-specific Issues

HP-UX Itanium

Applications using the Oracle Access Module on HP-UX 11i v3 crash with a bus error if the first connection established by the application is inside a child thread. Increasing the thread stack size using *thread_attr_setstacksize()* resolves the issue (SP-9006).

Red Hat Enterprise Linux, Intel C++ Compilers

Using the Intel C++ compiler may result in unresolved symbols errors. Supplying the path to the lib subdirectory of the compiler installation as the first path for the linker to search for libraries (for example: `-L<path to compiler>/lib`) resolves the errors. (SP-8484)

In addition, the path to the Intel libraries should also appear before the path to the Oracle libraries in the `LD_LIBRARY_PATH` environment variable. (SP-8585)

SourcePro Net

Secure Communications

- **Using SSLv2.**

Clients using SSLv2 may no longer be able to connect to servers using TLSv1WithFallback. This is because the default cipher list used by OpenSSL 1.0.0 and later no longer includes the necessary ciphers for this fallback to work. While it is strongly encouraged to avoid using SSLv2, fallback support for SSLv2 can be enabled by enabling the necessary ciphers on a context. The following code snippet demonstrates.

```
RWSecureSocketContext context;  
context.setCipherList("ALL:!NULL");
```

Windows - Visual Studio 2015

- **Access violation errors when using the OpenSSL DLL**

To avoid access violation errors when using the OpenSSL DLL, be sure that the application and the OpenSSL DLL link to the same Microsoft CRT. (SP-20754)

SourcePro Analysis

Essential Math Module

Windows - Intel C++

Intel's Math Kernel Library v.11.3 links libcpmt.lib. If SourcePro is built statically with the Intel compiler, you may see warnings that contain the message "'MT_StaticRelease' does not match value 'MD_DynamicRelease'". This has been reported to Intel and should be fixed in its next release of the MKL. You can suppress this warning with the linker option: `"/NODEFAULTLIB:libcpmt.lib"`.

Currency Module

- **Inexact exception for results with insignificant trailing zeros**

Multiplication where the result contains insignificant trailing zeros may cause an inexact exception to be thrown, even if the result is represented exactly (SP-10262). In the following example, in which the class *RWDecimal*<*RWMultiPrecisionInt*<2>> can represent numbers with 18 decimal digits, an inexact exception will be generated:

```
RWDecimal< RWMultiPrecisionInt<2> > valA = "5282.80977864";
RWDecimal< RWMultiPrecisionInt<2> > valB = "10000000000";
RWDecimal< RWMultiPrecisionInt<2> > valMul = valA*valB;
cout << "valA : " << valA << endl;
cout << "valB : " << valB << endl;
cout << "valMul: " << valMul << endl;
```

The program creates the following output:

```
[MULINEXACT] RWDecimal: inexact result in multiplication
valA : 5282.80977864
valB : 10000000000
valMul : 5282809778640.00000 // 18 digit result still exact
```

In one sense, the inexact exception is correct, since a certain number of significant digits are lost in the calculation. It just happens that those digits are all zeros.

Windows & Red Hat Linux - Intel C++

- **Floating Point Variables.**

Intel C++ compiler optimizations include storing floating point variables in registers. On some machines this results in greater precision than a double should have.

According to the Intel C++ compiler User's Guide:

```
-fp-model option or /fp:precise (Windows) Tells the compiler to strictly adhere to
value-safe optimization when implementing floating-point calculations. It disables
optimizations that can change the result of floating-point calculations, which is
required for strict ANSI conformance
```

This advice applies to portions of the Currency Module code.

We strongly recommend that you add `-fp-model precise` (Linux) or `/fp:precise` (Windows) to the command line of any non-debug programs built using the Currency Module and the Intel C++ compiler.

Red Hat/SUSE Linux - GCC

- **Floating Point Variables.**

GCC compiler optimizations include storing floating point variables in registers. On some machines this results in greater precision than a double should have.

According to the GCC man page:

"For most programs, the excess precision does only good, but a few programs rely on the precise definition of IEEE floating point. Use '-ffloat-store' for such programs."

This advice applies to portions of the Currency Module code, and we have disabled this particular optimization for non-debug library builds and example-program builds by adding the `-ffloat-store` option to the compiler command line.

We strongly recommend that you add `-ffloat-store` to the command line of any non-debug programs built using the Currency Module and the GCC compiler.

Documentation

- **Inter-document Hyperlinks do not Work in the PDF Documents.**

The PDF documents shipped with SourcePro are provided so customers can print the documents if they wish. These documents are not intended as primary access to the product documentation, and may not be as fully functional as the HTML documentation. One known issue is that hyperlinks between these documents do not work, and other issues may exist.

- **Windows 8 PDF Reader May Behave Inconsistently.**

The native Windows 8 PDF Reader may behave inconsistently; it has been observed that hypertext links are not processed correctly. Instead, use Adobe's free Acrobat Reader.



Rogue Wave provides software development tools for mission-critical applications. Our trusted solutions address the growing complexity of building great software and accelerates the value gained from code across the enterprise. Rogue Wave's portfolio of complementary, cross-platform tools helps developers quickly build applications for strategic software initiatives. With Rogue Wave, customers improve software quality and ensure code integrity, while shortening development cycle times.